UA/NPPS-1-04

A numerical algorithm for computing the complete set of one-gluon loop diagrams in QCD on the basis of a single master expression

A. S. Kapoyannis*, A. I. Karanikas and C. N. Ktorides

University of Athens, Physics Department

Nuclear & Particle Physics Section

Panepistimiopolis, Ilisia GR 157–71, Athens, Greece

Abstract

A numerical program is presented which facilitates the computation of the full set of one-gluon loop diagrams (including ghost loop contributions), with M attached external gluon lines in all possible ways. The feasibility of such a task rests on a suitably defined master formula, which is expressed in terms of a set of Grassmann and a set of Feynman parameters, the number of which increases with M. An important component of the numerical program is an algorithm for computing multi-Grassmann variable integrals. The cases M=2, 3, 4, which are the only ones having divergent terms, are fully worked out. A complete agreement with known, analytic results pertaining to the divergent terms is attained.

PACS: 02.70.Rw, 12.38.Bx

*Corresponding author. *E-mail address:* akapog@nikias.cc.uoa.gr (A. S. Kapoyannis)

PROGRAM SUMMARY

Title of program: DILOG

Program obtainable from: CPC Program Library, Queen's University of Belfast, N. Ireland

Computer for which the program is designed on and others on which it has been tested: Personal Computer

Operating systems or monitors under which the program has been tested: Windows 98

Programming language used: FORTRAN 77

Memory required to execute with typical data: 73 728 words

No. of bits in a word: 64

No. of processors used: one

Has the code been vectorized or parallelized?: no

No of bytes in distributed program, including test data, etc: 589 824

Keywords: Computation of one-gluon loop Feynman diagram in QCD

Nature of Physical problem

The computation of loop diagrams in QCD with many external gluon lines is a time consuming task, practically beyond reasonable reach of analytic procedures. In this paper we apply recently proposed master formulas for the computation of one-loop gluon diagrams in QCD with ah arbitrary number, M, external gluon lines.

Method of solution

The Grassmann variables in the master expressions and their properties are represented in suitably defined integer matrices, in order to carry out the Grassmnan integration first. The parametric functions are handled by an integer representation. The output of the program is the complete analytic result (tested for validity up to $M = 4$).

Restrictions on the complexity of the problem: M must one exceed 4, though extension to higher values is simple.

Typical running time: 25 seconds for $M = 4$

LONG WRITE-UP

1. Introduction

Quantum Chromodynamics (QCD) enjoys, in our days, universal acceptance as the fundamental theory for the strong interaction. As a quantum field theoretical system, QCD has been extensively applied to situations in which its perturbative content provides a dependable computational tool. It is, in fact, within the framework of this perturbative content that QCD has successfully confronted the quantitative description of the multitude of scattering processes, which probe strong interaction dynamics at high energies. Admittedly, the study of the non-perturbative domain of the theory offers intriguing challenges. Nevertheless, the immediate need to confront recent measurements coming from the HERA and Tevatron particle accelerators as well as the expected ones, in the near future, from the LHC accelerator continues to put perturbative QCD (pQCD) to the forefront of theoretical activity[1].

Given the non-abelian structure of QCD, the (by far) most demanding component of the theory in relation to perturbative calculations is its gluonic, as opposed to its quark, sector[2]. In particular, perturbative computations involving Feynman diagrams with gluon/ghost loops become, to say the least, quite monstrous. During the last decade or so various methods, aiming to expedite Feynman diagram computations in QCD, have been proposed whose basic feature is that they rely in a first, rather than the usual second, quantization approach to the formulation of the theory. Corresponding attempts have employed either strings [1-3], or world-line paths [4-10] as their underlying basic agents. Within the framework of the latter case, two of the present authors, Refs [9,10], were involved in work which led to the formulation of a set of master expressions, that condense the multitude of all Feynman diagrams entering a given configuration determined by the number of loops and the number of external propagators attached on them. To be more precise, the derived expressions go up to two loop configurations, nevertheless the 'logic' of the construction can be extended to loops of higher order. Even so, the analytical confrontation of a two loop situation with four 'external' gluon lines constitutes a challenging enough problem [11].

[1]Non-perturbative input in the relevant computations enters in the form of 'initial' phenomenological information, with pQCD taking charge from thereon.

[2]For that matter, this is more so the case for the non-perturbative domain of the theory.

The basic feature of the master expressions arrived at in Refs [9,10] is that they are furnished in terms of a set of Grassman and a set of Feynman variables. Once integrations over these two sets of variables are performed one obtains the full result, i.e. the one which, for the given configuration, contains the contribution of all Feynman diagrams at once. It is obvious, even before laying an eye on these master formulas, that in order to put them into practical use, one should employ suitable computational methods for confronting them. It is the aim of this paper to present such a program, which will be applied to the one gluon loop case for two, three and four external gluonic lines. Given that one part of our program deals with the confrontation of multi-Grassmann variable integrals, which are entangled with expressions involving additional variables (also in line to be integrated over), it is hoped that it could find applicability to other situations, where Grassmann variables also make their entrance.

Our paper is organized as follows. In the following section we present the battery of formulas, which are associated with the master expression corresponding to one gluon/ghost loop with M external gluon attachments in all possible ways. We intend to consider the cases $M = 2$, 3 and 4, which exhibit divergent terms[3], in addition to finite ones. In section 3 we describe the structure of the program, while section 4 presents our results, with the first two cases being explicitly displayed. Finally, our concluding remarks are made in section 6.

2. The one loop master formula

Consider a configuration consisting of one gluon/ghost loop onto which M external gluon lines, with corresponding momenta $p_1 \cdots p_M$ are attached (see Figure). According to Ref [9], the master expression, which summarizes the total contribution from all Feynman diagrams pertaining to this configuration is given by

$$\Gamma_1^{(M)}(p_1,\ldots,p_M) = -\frac{1}{2} g^M (2\pi)^4 \delta^{(4)}\left(\sum_{n=1}^{M} p_n\right) Tr_C(t_G^{\alpha_M}\cdots t_G^{\alpha_1}) \frac{1}{(4\pi)^2}\int_0^\infty dT T^{M-3}\times$$

$$\times\left[\prod_{n=M}^{1}\int_0^1 du_n\right]\theta(u_M,\ldots,u_1)F^{(M)}(u_1,\ldots,u_m;T)\exp\left[T\sum_{n<m} p_n\cdot p_m G(u_n,u_m)\right]$$

$$+permutations \qquad , \qquad (1)$$

[3] As expected, the aforementioned master expressions implicate the absence of divergent terms for $M > 4$, cf. Ref [9].

where g is the coupling constant of the theory, the $t_G^{\alpha_i}$, $i = 1, \ldots, M$ are the $SU(3)_{color}$ group generators (in the adjoint representation) with Tr_C the trace over the color group, the u_i are Feynman parameters, the function θ is specified by
$\theta(u_M, \ldots, u_1) = \theta(u_M - u_{M-1}) \ldots \theta(u_2 - u_1)$ and

$$F^{(M)}(u_1, \ldots, u_M; T) = \left[\prod_{n=M}^{1} \int d\xi_n d\bar{\xi}_n \right] \left(Tr_L \Phi^{[1]} - 2 \right) \times$$

$$\times \exp \left[\sum_{n \neq m} \xi_n \bar{\xi}_n \varepsilon^n \cdot p_m \partial_n G(u_n, u_m) + \frac{1}{2T} \sum_{n \neq m} \xi_n \bar{\xi}_n \xi_m \bar{\xi}_m \varepsilon^n \cdot \varepsilon^m \partial_n \partial_m G(u_n, u_m) \right] , \quad (2)$$

In the above equation ξ's are Grassmann variables, the ε^i are polarization vectors for the external gluons, $\Phi^{[1]}$ is the so-called *spin factor* entering the world-line description of QCD (see below), with Tr_L denoting trace with respect to Lorenz generator representation indices and the $G(u_n, u_m)$ are free propagators for the particle modes entering the worldline path integral description of QCD, in the context of its first quantized version (see Ref [9]), obeying the equation(s)

$$-\partial_n \partial_m G(u_n, u_m) = \partial_n^2 \dot{G}(u_n, u_m) \equiv \ddot{G}(u_n, u_m) = 2[\delta(u_n, u_m) - 1] , \quad (3)$$

with boundary condition

$$\partial_n G(u_n, u_m) \equiv \dot{G}(u_n, u_m) = sign(u_n - u_m) - 2(u_n - u_m) = -\dot{G}(u_m, u_n) , \quad (4)$$

The explicit expression for the spin factor in terms of the set of parameters entering our expressions is (the $J_{\mu\nu}$ are the Lorentz generators, in the vector representation)

$$\Phi^{[1]}_{\mu\nu} = P \exp \left[\frac{i}{2} \sum_{n=1}^{M} J \cdot \phi(n) \right]_{\mu\nu} =$$

$$= \delta_{\mu\nu} + \frac{i}{2} (J_{\rho\sigma})_{\mu\omega} \sum_{n=1}^{M} \phi_{\rho\sigma}(n) + (\frac{i}{2})^2 (J_{\rho_2\sigma_2})_{\mu\lambda} (J_{\rho_1\sigma_1})_{\lambda\nu} \sum_{n_2=1}^{M} \sum_{n_1=1}^{n_2-1} \phi_{\rho_2\sigma_2}(n_2) \phi_{\rho_1\sigma_1}(n_1) + \ldots , \quad (5)$$

where

$$\phi_{\mu\nu}(n) = 2\bar{\xi}_n \xi_n (\varepsilon_\mu^n p_{n,\nu} - \varepsilon_\nu^n p_{n,\mu}) + \frac{4}{T} \bar{\xi}_{n+1} \xi_{n+1} \bar{\xi}_n \xi_n (\varepsilon_\mu^{n+1} \varepsilon_\nu^n - \varepsilon_\nu^{n+1} \varepsilon_\mu^n) \delta(u_{n+1} - u_n) . \quad (6)$$

A point of note is the following: In the above expressions a specific time ordering has been chosen according to which index $n + 1$ comes immediately after index n, with $\xi_{M+1} = \bar{\xi}_{M+1} = 0$.

3. Presenting the program structure

The form of eq. (1) is considered as a sum of different elements. Every element is described with a line in three matrices: $C1$, $C2$ and $C3$.

Matrix $C1$ contains the Grassmann part of the form. It is a matrix with integer elements with dimensions $(NC1 \times NC2)$[4].

Matrix $C2$ has dimensions $(NC1 \times 1)$ and contains real numbers.

Matrix $C3$ represents the functions that accompany the Grassmann variables and its dimensions are $(NC1 \times NC3)$. The representation of the functions will be discussed in the "Functions Handling" subsection.

3a. Grassmann Coding

A Grassmann variable is represented in matrix $C1$ by an integer number. This integer is equal to the index of the Grassmann number for the ξ type variables and equal to the opposite of the index of the Grassmann number for the $\bar{\xi}$ type variables. So

$$\xi_n \;\rightarrow\; n\,, \quad \bar{\xi}_n \;\rightarrow\; -n\,. \tag{7}$$

A form that contains products of Grassmann variables is represented by a line in matrix $C1$. The elements in this line are put in the same order by which the Grassmann variables enter a given product. This line is accompanied by a line in matrix $C2$ which contains a real number and a line matrix $C3$ which represents the functions. For example the following coding would take place

$$\alpha \cdot func_1 \cdot func_2 \cdot \xi_1\bar{\xi}_1\xi_5\bar{\xi}_5 \;\rightarrow \overset{C2}{(\ \alpha\ )}\overset{C3}{(\ f_1 \quad f_2\ )}\overset{C1}{(\ 1 \quad -1 \quad 5 \quad -5\ )}\,, \tag{8}$$

where the integers f_1, f_2 represent the functions $func_1, func_2$ and will be discussed in the following subsection.

The addition of two forms containing Grassmann variables (represented as $\oplus$) is carried as follows

$$\alpha_1 \cdot func_1 \cdot func_2 \cdot \xi_2\bar{\xi}_2 + \alpha_2 \cdot func_3 \cdot \xi_3\bar{\xi}_3\xi_1\bar{\xi}_1 \rightarrow$$
$$\overset{C2}{(\ \alpha_1\ )}\overset{C3}{(\ f_1 \quad f_2\ )}\overset{C1}{(\ 2 \quad -2\ )} \oplus \overset{C2}{(\ \alpha_2\ )}\overset{C3}{(\ f_3\ )}\overset{C1}{(\ 3 \quad -3 \quad 1 \quad -1\ )} \rightarrow$$

[4]The first number is the number of lines and the second the number of columns.

$$\overset{C2}{\begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix}} \overset{C3}{\begin{pmatrix} f_1 & f_2 \\ f_3 & 0 \end{pmatrix}} \overset{C1}{\begin{pmatrix} 2 & -2 & 0 & 0 \\ 3 & -3 & 1 & -1 \end{pmatrix}} \ . \tag{9}$$

The result of the addition of two forms represented with two sets of matrices $C2$, $C3$ and $C1$ will be a new set of these matrices with new dimensions. If the dimensions of the first set of matrices are represented by N's and the these of the second set by M's, then the resulting dimensions of the final set will be

$$\overset{C2}{(N_1 \times 1)}\overset{C3}{(N_1 \times N_3)}\overset{C1}{(N_1 \times N_2)} \oplus \overset{C2}{(M_1 \times 1)}\overset{C3}{(M_1 \times M_3)}\overset{C1}{(M_1 \times M_2)} \rightarrow$$

$$\overset{C2}{[(N_1 + M_1) \times 1]}\overset{C3}{[(N_1 + M_1) \times \max(N_3, M_3)]}\overset{C1}{[(N_1 + M_1) \times \max(N_2, M_2)]} \ . \tag{10}$$

So the dimensions of the matrices which represent our forms is not constant and it is important for every part of the program to be known. Where it is necessary, the lines of matrices $C3$ and $C1$ are filled with zeros. The addition of two forms is carried out in the program by the subroutine "ADD".

The multiplication of two forms containing Grassmann variables (represented as $\otimes$) is carried as follows

$$\left(\alpha_1 \cdot func_1 \cdot func_2 \cdot \xi_2\bar{\xi}_2 + \alpha_2 \cdot func_3 \cdot \xi_3\bar{\xi}_3\xi_1\bar{\xi}_1\right) \cdot \left(\beta_1 \cdot func_4 \cdot \xi_5\bar{\xi}_5 + \beta_2 \cdot func_5 \cdot \xi_7\bar{\xi}_7\right) \rightarrow$$

$$\overset{C2}{\begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix}} \overset{C3}{\begin{pmatrix} f_1 & f_2 \\ f_3 & 0 \end{pmatrix}} \overset{C1}{\begin{pmatrix} 2 & -2 & 0 & 0 \\ 3 & -3 & 1 & -1 \end{pmatrix}} \otimes \overset{C2}{\begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix}} \overset{C3}{\begin{pmatrix} f_4 \\ f_5 \end{pmatrix}} \overset{C1}{\begin{pmatrix} 5 & -5 \\ 7 & -7 \end{pmatrix}} \rightarrow$$

$$\overset{C2}{\begin{pmatrix} \alpha_1 \cdot \beta_1 \\ \alpha_1 \cdot \beta_2 \\ \alpha_2 \cdot \beta_1 \\ \alpha_2 \cdot \beta_2 \end{pmatrix}} \overset{C3}{\begin{pmatrix} f_1 & f_2 & f_4 \\ f_1 & f_2 & f_5 \\ f_3 & 0 & f_4 \\ f_3 & 0 & f_5 \end{pmatrix}} \overset{C1}{\begin{pmatrix} 2 & -2 & 0 & 0 & 5 & -5 \\ 2 & -2 & 0 & 0 & 7 & -7 \\ 3 & -3 & 1 & -1 & 5 & -5 \\ 3 & -3 & 1 & -1 & 7 & -7 \end{pmatrix}} \ . \tag{11}$$

The result of the multiplication of two forms represented with two sets of matrices $C2$, $C3$ and $C1$ will be a new set of these matrices with new dimensions. If the dimensions of the first set of matrices are represented by N's and those of the second set by M's, then the resulting dimensions of the final set will be

$$\overset{C2}{(N_1 \times 1)}\overset{C3}{(N_1 \times N_3)}\overset{C1}{(N_1 \times N_2)} \otimes \overset{C2}{(M_1 \times 1)}\overset{C3}{(M_1 \times M_3)}\overset{C1}{(M_1 \times M_2)} \rightarrow$$

$$[(N_1 \cdot \overset{C2}{M_1}) \times 1][(N_1 \cdot M_1) \overset{C3}{\times} (N_3 + M_3)][(N_1 \cdot M_1) \overset{C1}{\times} (N_2 + M_2)] \ . \tag{12}$$

The multiplication of two forms is carried out in the program by the subroutine "MULTIPLY".

Certain simplifications and rearrangements are carried out in the Grassmann and Function Number matrices upon the completion of certain routines. This is done by subroutine "REARRANGE" which performs the following acts:

• It checks if in the same line two equal Grassmann variables exist and if that happens it puts zeros everywhere in the lines of matrices $C2$, $C3$ and $C1$. This accounts for the fact that ξ_n^2=$\bar{\xi}_n^2$=0.

• If a zero exists in one line of matrix $C2$, then the corresponding lines in all matrices are omitted, all the following lines are moved in front by one line and the line dimension is reduced by one.

• It collects the Grassmann variables to the left of every line of matrix $C1$, ignoring the zeros in between. If this reduces the dimension of the columns of matrix $C1$, it performs the reduction. The same act is performed on function matrix $C3$.

• It places the integers which represent the Grassmann variables in every line of matrix $C1$ in ascending order, according to their absolute value. Among two variables with the same absolute value the positive is placed to the left of the negative one. The sign of the corresponding value of matrix $C2$ is changed according to the changes made to the order of the Grassmann variables.

• It places the number which represents the functions in every line of matrix $C3$ in ascending order. The set of functions which are not allowed to change their relative order are placed to the left. Functions represented by numbers $3nm$ with $n > m$ are entered as $3mn$. This practice, as well as the two preceding ones, are needed for comparing the different lines of the matrices.

• If two lines in the matrix $C1$ and the corresponding lines in the matrix $C3$ are found equal, then the accompanying factors of matrix $C2$ are added. The second line is omitted in all matrices and their line dimension is reduced by one.

• The lines of the matrices are also rearranged. The lines with all elements of matrices

$C1$ and $C3$ equal to zero are entered first.

The matrices $C3$ and $C1$ are two-dimensional, but their dimension is not known *a priori*. This can cause problems when the matrices are fed from one subroutine to another. In every subroutine the dimensions of these matrices are pre-defined so that they will always be larger than what is required. Also, it is not always possible to define the same dimensions for these matrices in all the subroutines. The FORTRAN language stores the elements of a two-dimensional matrix in neighbouring memory places so that the first column is stored first, then the second, etc. When these places are read from another subroutine they will not result to the same matrix in the new subroutine, unless the matrix has the same dimensions. To avoid this problem the subroutine "LINEUP" is used, which rearranges the elements of a matrix that is defined with dimensions $(N_1 \times N_2)$ so that they are stored exactly the same way as they would be stored if the dimensions of the matrix were $(N_1' \times N_2')$. In other cases we deal with this problem by using one-dimensional arrays to store two-dimensional matrices. The elements of the matrix a_{ij}, with dimensions $(N_1 \times N_2)$, are stored into the elements b of the array, so that the first column of the matrix is stored into the first elements of the array, then the second, etc., according to the correspondence

$$a_{ij} \quad \rightarrow \quad b_{(j-1)\cdot N_1+i} \ . \tag{13}$$

To evaluate eq. (1) we first code the following form in matrices $C2$, $C3$ and $C1$

$$T \sum_{n<m} p_n \cdot p_m G(u_n, u_m) + \sum_{n \neq m} \xi_n \bar{\xi}_n \varepsilon^n \cdot p_m \partial_n G(u_n, u_m) +$$

$$\frac{1}{2T} \sum_{n \neq m} \xi_n \bar{\xi}_n \xi_m \bar{\xi}_m \varepsilon^n \cdot \varepsilon^m \partial_n \partial_m G(u_n, u_m) \delta(u_{n+1} - u_n) \ . \tag{14}$$

This is done by subroutine "KNKM". Then the exponent of the above form has to be evaluated. For this purpose the subroutine "EXPONENTIAL" has been constructed. In this routine it is checked whether a part that does not contain Grassmann variables exists. If this is the case, the exponent of this part will only multiply all the lines of the rest. Then the exponent of the Grassmann containing part is calculated. This exponent will only contain M terms, since there are only M different Grassmann pairs $\xi_n \bar{\xi}_n$. Every term is constructed by multiplying the Grassmann part of "KNKM" as many times as it is needed.

Every newly constructed term is then added to the pervious ones. The result is stored to the output of subroutine "EXPKNKM".

The structure $(i/2)J \cdot \phi(n)$ is coded in subroutine "FMN". The index $n+1$ is evaluated from the specific time order which has been chosen. The exponent of "FMN" is found by subroutine "PEXPONENTIAL". The trace of $\delta_{\mu\nu}$ (the first term in the sum of eq. (6)) is accounted for by putting the factor 4 to a line of matrix $C2$.

The complete result is determined in subroutine "MULTIPLYALL". In the output of "PEXPONENTIAL" a line with zeros in the domain of matrices $C3$ and $C1$ but with the factor -2 in $C2$ is added, to construct the form $Tr_L\Phi^{[1]} - 2$. Then the result is multiplied with the result of "EXPKNKM". Now the complete Grassmann part of the calculation is stored to matrix $C1$. The Grassmann integration is simple, since it amounts to keeping only the terms where all the Grassmann variables are present. Another simplification is made by neglecting the terms that contain only one function to be traced[5] and which has zero trace. Our calculation at the end of "MULTIPLYALL" is stored only in matrices $C2$ and $C3$ since the Grassmann variables have been integrated out.

3b. Function Handling

In order to keep the matrices that represent our forms as small as possible, we have first carried out the Grassmann integrations in order to remove completely the Grassmann variables. In fact, this is the reason we have coded the functions that accompany the Grassmann variables in matrix $C3$. Every line of this matrix contains a product of functions which are represented by integer numbers according to the labelling described in Table 1.

In order for the representation of Table 1 to be valid it must be $M \leq 9$. The asterisk has the meaning that the corresponding functions are considered non-commutative and the order that they have been inserted is not allowed to change. Their Lorentz indices μ, ν are also considered indefinite, that is they are not defined at this point by the number which represents the function. So when in a line of matrix $C3$ functions of the type $2n$ or $3n$ are found together it is assumed that the Lorentz trace has to be performed, according to the exact order the functions are found in the respective line. For example if the numbers 21,

[5]Function of the type $2n$ or $3n$ of the next subsection.

22 and 31 are found in one line of $C3$ (placed in this order and without any other function of the type $2n$ or $3n$ in the same line), then it is assumed that

$$21\ 22\ 31\ \rightarrow\ (21)_{\mu\rho}\ (22)_{\rho\sigma}\ (31)_{\sigma\mu}\ \rightarrow$$

$$(i/2)(J_{\alpha\beta})_{\mu\rho}2(\varepsilon^1_\alpha p_{1,\beta}-\varepsilon^1_\beta p_{1,\alpha})(i/2)(J_{\gamma\delta})_{\rho\sigma}2(\varepsilon^2_\gamma p_{2,\delta}-\varepsilon^2_\delta p_{2,\gamma})$$
$$(i/2)(J_{\kappa\lambda})_{\sigma\mu}(1/T)(\varepsilon^2_\kappa\varepsilon^1_\lambda-\varepsilon^2_\lambda\varepsilon^1_\kappa) \tag{15}$$

In this way we postpone the evaluation of the trace until after the Grassmann integration.

After the Grassmann integration we are able to perform the trace in the surviving terms. This is accomplished by subroutine "TRACE". The information contained in matrix $C3$ is divided into two matrices $C3$ and $C6$. The new matrix $C3$ contains only the following functions according to the labelling

Matrix $C6$ contains in its first column the exponent of T followed by internal products of the vectors ε's and p's, according to the coding of Table 1.

In order to arrive at that output, the maximum number of traces to be calculated is evaluated first, so that the dimension of the matrix that will contain the result is determined. The saturation of indices ρ, σ in $J_{\rho\sigma}$ is performed instantly, since

$$(i/2)(J_{\rho\sigma})_{\mu\nu}2(\varepsilon^n_\rho p_{n,\sigma}-\varepsilon^n_\sigma p_{n,\rho})=-2(\varepsilon^n_\mu p_{n,\nu}-\varepsilon^n_\nu p_{n,\mu})\ , \tag{16}$$

$$(i/2)(J_{\rho\sigma})_{\mu\nu}(1/T)(\varepsilon^{n+1}_\rho\varepsilon^n_\sigma-\varepsilon^{n+1}_\sigma\varepsilon^n_\rho)=-(1/T)(\varepsilon^{n+1}_\mu\varepsilon^n_\nu-\varepsilon^{n+1}_\nu\varepsilon^n_\mu) \tag{17}$$

Then the indefinite Lorentz indices become definite. We begin by giving the indices 1,2 to the first function that enters the product to be traced, then the indices 2,3 to the second, etc, until the last one which takes as second index 1. The representation used is

$$\varepsilon^n_\nu\ \rightarrow\ 3n\nu\ ,\quad p_{n,\nu}\rightarrow\ 2n\nu\ . \tag{18}$$

The products are then represented as lines in two temporary matrices and every term in a sum uses a separate line, for example

$$\varepsilon^1_1 p_{2,2}-\varepsilon^1_2 p_{2,1}\ \rightarrow \begin{pmatrix} 1 \\ -1 \end{pmatrix}\begin{pmatrix} 311 & 222 \\ 312 & 221 \end{pmatrix}\ . \tag{19}$$

The addition and the multiplication of forms as the above is carried out in the same way as was done when we dealt with the Grassmann variables, until all the terms to enter the product to be traced are exhausted. In order to form internal products of the different elements, the last digit of every element is compared to the last digit of all the elements to the right in the same line. When two last digits are found equal an internal product between the corresponding elements is formed and represented according to the prescription of Table 3.

We have, also, taken into account the rule $\varepsilon^n \cdot p_n = 0$ and we have made everywhere in matrix $C6$ (but not in $C3$) the replacement $p_1 = -p_2 - p_3 \ldots - p_M$, according to momentum conservation.

The result that corresponds to one line of matrix $C3$ is put in one line of matrix $C6$. This contains a sum of different terms, each of which is a product of internal products accompanied by a factor. When the line of matrix $C6$ is read, an integer less or equal to 2000 (which represents a factor) signals the beginning of the product of internal products and a another one signals its ending. For example the following translation is assumed from a line of matrix $C6$

$$-1 \quad 3431 \quad 1 \quad 3433 \quad 3223 \quad 3122 \quad 2424 \quad -3 \quad 3423 \quad 3332 \quad \rightarrow$$

$$-\varepsilon^4 \cdot \varepsilon^1 + \varepsilon^4 \cdot \varepsilon^3 \; \varepsilon^2 \cdot p_3 \; \varepsilon^1 \cdot p_2 \; p_4 \cdot p_4 - 3 \; \varepsilon^4 \cdot p_3 \; \varepsilon^3 \cdot \varepsilon^2 \; . \tag{20}$$

4. Results

At the beginning of its run the program asks the user to insert the number of external gluons M. Then it calculates all the possible combinations of time order and lists them for the user to see. Every combination is accompanied by an integer number and the user is expected to insert one of these numbers in order to choose the desirable combination.

Then the calculation is carried out. Several files are used during this calculation. The results after the Grassmann integration (matrices $C2$ and $C3$ in coding of Table 1) are written in file "TEMP2". The final result (matrices $C2$, $C3$ in coding of Table 2 and $C6$ in coding of integers) are written in file "TEMP4". The result in function like output is written to file "TEMP5". The integers NC1, NC3 and NC6 are connected to the dimensions of the output matrices. These are $C2(NC1 \times 1)$, $C3(NC1 \times NC3)$ and $C6(NC1 \times NC6)$. The

result is read as a sum of different terms. Every term is formed by reading a line of matrix $C2$, then the corresponding line of matrix $C3$ and then the corresponding line of matrix $C6$. When a lot of non-zero elements exist in some lines of matrix $C6$, these lines are folded beneath in order to permit easy reading. Then the whole sum is supposed to be multiplied by the function and integrated by the multiple integral which appear at the begging of the output.

The number of terms at the output ($NC1$) grows rapidly when M increases. For $M = 2$, three terms exist (shown to Table 4a). For $M = 3$, 23 terms exist. These terms are shown in Table 4b. For $M = 4$ the existing terms are 233. These include both finite and divergent terms.

Our results have been compared with the analytic calculations for the divergent terms for $M = 2$, $M = 3$ and $M = 4$ [9] and have been found in total agreement. The structure of the program and the methods used for the computation of the master formulas permit the extension of the calculations for value of M higher than 4.

5. Concluding remarks

In this paper a computational algorithm has been presented for the successful computation of the complete set of one-gluon loop Feynman diagrams with two, three and four external gluon attachments, on the basis of the master formulas derived in Ref [9]. It is hoped that the particular feature of the constructed algorithm, namely the ability to expedite integrations over a multivariable set, a subset of which is Grassmannian, could find wider applications to analogous situations that may arise in other physical problems wherein Grassmann variables make their entrance. Within the context of the present application, it would be of interest to apply the particular algorithm developed in this work to the two-gluon loop $M = 4$ case, the corresponding master expressions for which have been derived in Ref [10]. As a first attempt, one could restrict the relevant computation to the divergent term associated with the $M = 2$ configuration and verify the consistency with second order corrections to the running coupling constant in pQCD.

Acknowledgement

Two of the present authors, A. I. K. and C. N. K. acknowledge the support from the

General Secretariat of Research and Technology of the University of Athens.

References

[1] Z. Bern and D. A. Kosower, Nucl. Phys. B **321** (1989)605; B **379** (1992)451.

[2] Z. Bern, L. J. Dixon and D. A. Kosower, Phys. Rev. Lett. **70** (1993)2677 [arXiv:hep-ph/9302280].

[3] P. Di Vecchia, L. Magnea, A. Lerda, R. Russo and R. Marotta, Nucl. Phys. B **469** (1996)235 [arXiv:hep-th/9601143].

[4] M. J. Strassler, Nucl. Phys. B **385** (1992)145 [arXiv:hep-ph/9205205].

[5] M. Reuter, M. G. Schmidt and C. Schubert, Annals Phys. **259** (1997)313 [arXiv:hep-th/9610191].

[6] H-T. Sato and M. G. Schmidt, Nucl. Phys. B **524** (1998)742; it ibid. **560** (1999)551.

[7] H-T. Sato, M. G. Schmidt and Claus Zahlten, Nucl. Phys. B **579** (2000)492 [arXiv:hep-th/0003070].

[8] A. I. Karanikas and C. N. Ktorides, JHEP **9911** (1999)033 [arXiv:hep-th/9905027].

[9] S. D. Avramis, A. I. Karanikas and C. N. Ktorides, Phys. Rev. D **66** (2002)045017 [arXiv:hep-th/0205272].

[10] A. I. Karanikas and C. N. Ktorides, Phys. Lett. B **560**, 252 (2003) [arXiv:hep-th/0301214].

[11] Z. Bern, L. Dixon and D. A. Kosower, JHEP01 (2000)027.

Tables

Function	Number representing Function
$(i/2)(J_{\rho\sigma})_{\mu\nu}2(\varepsilon^n_\rho p_{n,\sigma} - \varepsilon^n_\sigma p_{n,\rho})$	2n (*)
$(i/2)(J_{\rho\sigma})_{\mu\nu}(1/T)(\varepsilon^{n+1}_\rho \varepsilon^n_\sigma - \varepsilon^{n+1}_\sigma \varepsilon^n_\rho)\delta(u_{n+1} - u_n)$	3n (*)
$\exp\{T\sum_{n=1}^{M}\sum_{m=n+1}^{M}[p_n \cdot p_m G(u_n, u_m)]\}$	100
$\varepsilon^n \cdot p_m\, \partial_n G(u_n, u_m)$	2nm
$(1/2T)\varepsilon^n \cdot \varepsilon^m \partial_n \partial_m G(u_n, u_m)$	3nm=3mn

Table 1. The representation of functions through integers before the Grassmann integration in matrix $C3$.

Function	Number representing Function
$\delta(u^{n+1} - u^n)$	3n
$\exp\{T\sum_{n=1}^{M}\sum_{m=n+1}^{M}[p_n \cdot p_m G(u_n, u_m)]\}$	100
$\varepsilon^n \cdot p_m\, \partial_n G(u_n, u_m)$	2nm
$\varepsilon^n \cdot \varepsilon^m \partial_n \partial_m G(u_n, u_m)$	3nm=3mn

Table 2. The representation of functions through integers in matrix $C3$ at the output.

Function	Number representing Function
$p_n \cdot p_m$	2n2m
$\varepsilon^n \cdot p_m$	3n2m
$\varepsilon^n \cdot \varepsilon^m$	3n3m

Table 3. The representation of functions through integers in matrix $C6$ at the output.

```
M= 2
NC1,NC3,NC6= 3 3 4
TIME ORDER= 1 2

-(pi**2/2)*g**(2)*dl4(p1+p2 )*TrC(tGa2 tGa1 )*

  inf          1            u2
   |    dT     |    du2     |     du1     th(u2,u1 )
   0           0            0


 4.0   EXP(all)                                          T**(-1)   +2    e2.e1    p2.p2
 2.0   EXP(all)       e1.p2d1G(u1,u2)    e2.p1d2G(u2,u1) T**(-1)
 2.0   EXP(all)    e1.e2d1d2G(u1,u2)                     T**(-2)
```

Table 4a. The function like output of the program for $M = 2$ and for the time order $u_1 < u_2$. The Memoradum is similar to Table 4b.

M= 3
NC1,NC3,NC6= 23 4 41
TIME ORDER= 1 2 3

-(pi**2/2)*g**(3)*dl4(p1+p2+p3)*TrC(tGa3 tGa2 tGa1)*

inf		1		u3		u2		
|	dT	|	du3	|	du2	|	du1	th(u3,u2,u1)
0		0		0		0		

-8.0	EXP(all)					T**(0)	-1	e3.p2	e2.p3	e1.p2		
							+1	e3.e2	e1.p2	p3.p2		
							+1	e3.e2	e1.p2	p3.p3		
							-1	e3.p2	e2.e1	p3.p3		
							+1	e3.e1	e2.p3	p2.p2		
							+1	e3.e1	e2.p3	p2.p3		
							-1	e3.e2	e1.p3	p2.p2		
							-1	e3.e2	e1.p3	p2.p3		
							-1	e3.e1	e2.p3	p3.p2		
							+1	e3.p2	e2.p3	e1.p3		
4.0	EXP(all)	e1.p2d1G(u1,u2)				T**(0)	+2	e3.p2	e2.p3	-2	e3.e2	p3.p2
4.0	EXP(all)	e1.p3d1G(u1,u3)				T**(0)	+2	e3.p2	e2.p3	-2	e3.e2	p3.p2
4.0	EXP(all)	e2.p1d2G(u2,u1)				T**(0)	-2	e3.p2	e1.p3	+2	e3.e1	p3.p2
							+2	e3.e1	p3.p3			
4.0	EXP(all)	e2.p3d2G(u2,u3)				T**(0)	-2	e3.p2	e1.p3	+2	e3.e1	p3.p2
							+2	e3.e1	p3.p3			
4.0	EXP(all)	e3.p1d3G(u3,u1)				T**(0)	-2	e2.p3	e1.p2	+2	e2.e1	p2.p2
							+2	e2.e1	p2.p3			
4.0	EXP(all)	e3.p2d3G(u3,u2)				T**(0)	-2	e2.p3	e1.p2	+2	e2.e1	p2.p2
							+2	e2.e1	p2.p3			
2.0	EXP(all)	e1.p2d1G(u1,u2)		e2.p1d2G(u2,u1)	e3.p1d3G(u3,u1)	T**(0)						
2.0	EXP(all)	e1.p2d1G(u1,u2)		e2.p1d2G(u2,u1)	e3.p2d3G(u3,u2)	T**(0)						
2.0	EXP(all)	e1.p2d1G(u1,u2)		e2.p3d2G(u2,u3)	e3.p1d3G(u3,u1)	T**(0)						
2.0	EXP(all)	e1.p2d1G(u1,u2)		e2.p3d2G(u2,u3)	e3.p2d3G(u3,u2)	T**(0)						
2.0	EXP(all)	e1.p3d1G(u1,u3)		e2.p1d2G(u2,u1)	e3.p1d3G(u3,u1)	T**(0)						
2.0	EXP(all)	e1.p3d1G(u1,u3)		e2.p1d2G(u2,u1)	e3.p2d3G(u3,u2)	T**(0)						
2.0	EXP(all)	e1.p3d1G(u1,u3)		e2.p3d2G(u2,u3)	e3.p1d3G(u3,u1)	T**(0)						
2.0	EXP(all)	e1.p3d1G(u1,u3)		e2.p3d2G(u2,u3)	e3.p2d3G(u3,u2)	T**(0)						
8.0	EXP(all)	dl(u3-u2)				T**(-1)	-2	e3.p2	e2.e1	+2	e3.e1	e2.p3
8.0	EXP(all)	dl(u2-u1)				T**(-1)	+2	e3.e1	e2.p3	-2	e3.e2	e1.p3
2.0	EXP(all)	e1.p2d1G(u1,u2)		e2.e3d2d3G(u2,u3)		T**(-1)						
2.0	EXP(all)	e1.p3d1G(u1,u3)		e2.e3d2d3G(u2,u3)		T**(-1)						
2.0	EXP(all)	e2.p1d2G(u2,u1)		e1.e3d1d3G(u1,u3)		T**(-1)						
2.0	EXP(all)	e2.p3d2G(u2,u3)		e1.e3d1d3G(u1,u3)		T**(-1)						
2.0	EXP(all)	e3.p1d3G(u3,u1)		e1.e2d1d2G(u1,u2)		T**(-1)						
2.0	EXP(all)	e3.p2d3G(u3,u2)		e1.e2d1d2G(u1,u2)		T**(-1)						

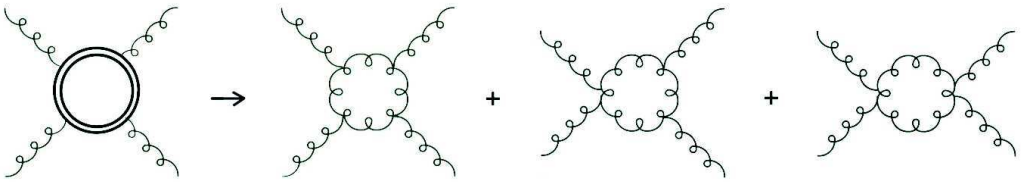
Memorandum:

pi**2	π^2	th(u3,u2,u1)	$\theta(u_3,u_2,u_1)$	e1.p2	$\varepsilon^1\cdot p_2$
dl4(p1+p2+p3)	$\delta^{(4)}(p_1+p_2+p_3)$	EXP(all)	$\exp\{T\sum_{n=1}^{M}\sum_{m=n+1}^{M}[p_n\cdot p_m G(u_n,u_m)]\}$	e2.e3	$\varepsilon^2\cdot\varepsilon^3$
TrC(tGa3 tGa2 tGa1)	$Tr_C(t_G^{\alpha_3}t_G^{\alpha_2}t_G^{\alpha_1})$	d2d3G(u2,u3)	$\partial_2\partial_3 G(u_2,u_3)$	d1G(u1,u2)	$\partial_1 G(u_1,u_2)$
inf	∞	dl(u3-u2)	$\delta(u_3-u_2)$	T**(-1)	T^{-1}

Table 4b. The function like output of the program for $M=3$ and for the time order $u_1<u_2<u_3$.

Figure Caption

Figure Illustration, for $M = 4$, of the classes of one-gluon loop Feynman diagram (right side of arrow) which are simultaneously accommodated by the corresponding master formula depicted on left side of arrow.


(a)
(b)
(c)